Task-Aware Image Downscaling

Heewon Kim, Myungsub Choi, Bee Lim, and Kyoung Mu Lee

Department of ECE, ASRI, Seoul National University, Seoul, Korea
{ghimhw,cms6539,biya999,kyoungmu}@snu.ac.kr
https://cv.snu.ac.kr

Abstract. Image downscaling is one of the most classical problems in computer vision that aims to preserve the visual appearance of the original image when it is resized to a smaller scale. Upscaling a small image back to its original size is a difficult and ill-posed problem due to information loss that arises in the downscaling process. In this paper, we present a novel technique called *task-aware image downscaling* to support an upscaling task. We propose an auto-encoder-based framework that enables joint learning of the downscaling network and the upscaling network to maximize the restoration performance. Our framework is efficient, and it can be generalized to handle an arbitrary image resizing operation. Experimental results show that our task-aware downscaled images greatly improve the performance of the existing state-of-the-art super-resolution methods. In addition, realistic images can be recovered by recursively applying our scaling model up to an extreme scaling factor of x128. We also validate our model's generalization capability by applying it to the task of image colorization.

Keywords: image downscaling, image super-resolution, deep learning

1 Introduction

Scaling or resizing is one of the most frequently used operations when handling digital images. When sharing images via the Internet, we rarely use the original high-resolution (HR) images because of the low resolution of display screens; most images are downscaled to save the data transfer cost while maintaining adequate image qualities. However, the loss of information from the downscaling process makes the inverse problem of super-resolution (SR) highly ill-posed, and zooming in to a part of the downscaled image usually shows a blurry restoration.

Previous works normally consider downscaling and super-resolution (upscaling) as separate problems. Studies on image downscaling [16,23,24,34] only focus on obtaining visually pleasing low-resolution (LR) images. Likewise, recent studies on SR [5,7,13,18,20,22,31,36,37] tend to fix the downscaling kernel (to *e.g.* bicubic downscaling) and optimize the restoration performance of the HR images with the given training LR-HR image pairs. However, the predetermined downscaling kernel may not be optimal for the SR task. Figure 1 shows an example of the importance of choosing an appropriate downscaling method, where the downscaled LR images in blue and red look similar, but the restored HR image

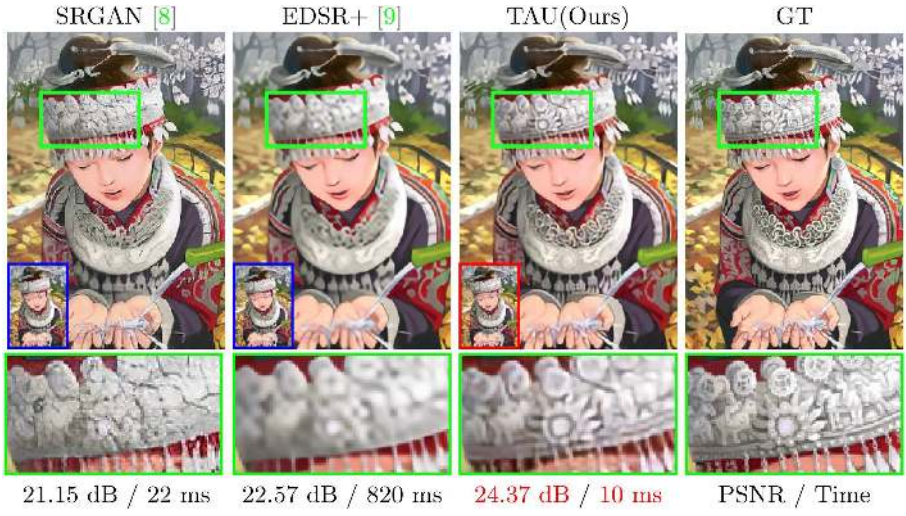

Fig. 1: Our task-aware downscaled (TAD) image (red box) generates more realistic and accurate HR image compared with the state-of-the-art methods that use bicubic-downscaled LR images (blue box). TAD image shows good LR image quality and, when upscaled with our jointly trained upscaling method TAU, outperforms EDSR+ by a large margin with considerably faster runtime. The scaling factor[1] is $\times 4$.

from the red LR image shows much more accurate result where the shapes and details are consistent with the original ground truth image.

In this paper, we address the problem of task-aware image downscaling and show the importance of learning the optimal image downscaling method for the target tasks. For the SR task, the goal is to find the optimal LR image that maximizes the restoration performance of the HR image. To achieve this goal, we use a deep convolutional auto-encoder model where the encoder is the downscaling network and the decoder is the upscaling network. The auto-encoder is trained end-to-end, and the output of the encoder (output of the downscaling network) will be our final task-aware downscaled (TAD) image. We also guarantee that the latent representation of the auto-encoder resembles the downscaled version of its original input image by introducing the *guidance image*. In SR, the guidance image is an LR image made by a predefined downscaling algorithm (*e.g.* bicubic, Lanczos), and it can be used to control the trade-off between HR image reconstruction performance and LR image quality. Our whole framework has only 20 convolution layers and can be run in real-time.

[1] We use the term *scaling factor* (denoted as sc) as "upscaling" factor unless otherwise mentioned. Then, downscaling an image from $H \times W$ to $\frac{H}{2} \times \frac{W}{2}$ is noted to have a scaling factor of $sc = \frac{1}{2}$. When indicated in a joint model, the images are downscaled to $\frac{1}{sc}$ and upscaled again to $\frac{sc}{sc} = 1$.

Our framework can also be generalized to other resizing tasks aside from SR. Note that the rescaling can be done not only in the spatial dimension but also in the channel dimension of an image. So we can apply our proposed framework to the grayscale-color conversion problem. In this setting, the downscaling task becomes RGB to grayscale conversion, and the upscaling task becomes image colorization. Our final grayscale image achieves visually much more pleasing results when re-colorized.

Overall, our contributions are as follows:

- To the best of our knowledge, our proposed method is the first deep learning-based image downscaling method that is jointly learned to boost the accuracy of an upscaling task. Applying our TAD images to train an SR model improves the reconstruction performance of the previous state-of-the-art (SotA) by a large margin.
- Our downscaling and upscaling networks operate efficiently and cover multiple scaling factors. In particular, our method achieves the best SR performance in extreme scaling factors up to $\times 128$.
- Our framework can be generalized to various computer vision tasks with scale changes in any dimension.

2 Related Work

In this section, we review studies on super-resolution and image downscaling.

2.1 Image Super-Resolution (SR)

Single image super-resolution (SR) is a standard inverse problem in computer vision with a long history. Most previous works discuss which methodology is used to obtain HR images from LR images, but we categorize SR methods according to the inherent assumptions they used with regard to the process of acquiring LR images in the first place. First, approaches without any such assumptions at all exist. These approaches include early methods that use interpolation [2, 12, 19, 39], which estimates filter kernels from local pixels/patch to the HR image pixel values with respect to the scaling factor. Interpolation-based methods are typically fast but yield blurry results. Many methods used priors from natural image statistics for more realistic textures [14, 28, 29]. One exceptional case of Ulyanov *et al.* [32] showed that a different structural image prior is inherent in deep CNN architecture.

Second, a line of work attempts to estimate the LR image acquisition process via self-similarities. These studies assume the fractal structures inherent in images, which means that considerable internal path redundancies exist within a single image. Glasner *et al.* [7] proposed a novel SR framework that exploits recurrent patches within and across image scales. Michaeli and Irani [22] improved this approach by jointly estimating the unknown downscaling blur kernel with the HR image, and Huang *et al.* [10] extended this approach to incorporate transformed self-exemplars for added expressive power. Shocher *et al.* [27]

recently proposed a "zero-shot" SR (ZSSR) using deep learning, which trains an image-specific CNN with HR-LR pairs of patches extracted from the test image itself. ZSSR shares our motivation of handling the problem of fixed downscaling process in generating HR-LR pairs when training deep models. However, the main objective is different in that our model focuses on restoring HR images *from previously downscaled images*.

The third and last category includes the majority of SR methods, wherein the process of obtaining LR images is predetermined (in most cases, MATLAB bicubic). Fixing the downscaling method is inevitable when creating a large HR-LR paired image dataset, especially when training a model needs a vast amount of data. Many advanced works that use neighbor embedding [3, 4, 6, 25, 31, 37], sparse coding [31, 35–37], and deep learning [5, 13, 17, 18, 20, 30] fall into this category, where many HR-LR paired patches are needed to learn the mapping function between them. With regard to more recent deep learning based methods, Dong *et al.* [5] proposed SRCNN as the first attempt to solve the SR problem with CNN. Accordingly, CNN-based SR architectures expanded, and they have greatly boosted the performance. Kim *et al.* (VDSR) [13] suggested the concept of residual learning to ease the difficulty in optimization, which was later improved by Ledig *et al.* (SRResNet) [18] with intermediate residual connections [8]. Following this line of work, Lim *et al.* [20] proposed an enhanced model called EDSR, which achieved SotA performance in the recent NTIRE challenge [30]. Ledig *et al.* proposed another distinctive method called SRGAN, which introduces adversarial loss with perceptual loss [11] and raised the problem of the current metric that we use for evaluating SR methods: peak signal-to-noise ratio (PSNR). Although these methods generate visually more realistic images than previous works regardless of their PSNR value, the generated textures can differ considerably from the original HR image (as shown in Figure 1).

2.2 Image Downscaling

Image downscaling aims to preserve the appearance of HR images in LR images. Conventional methods use smoothing filters and resampling for anti-aliasing [23]. Although these classical methods are still dominant in practical usage, more recent approaches have also attempted to improve the sharpness of LR images. Kopf *et al.* [16] proposed a content-adaptive method, wherein filter kernel coefficients are adapted with respect to image content. Öztireli and Gross [24] proposed an optimization framework to minimize SSIM [33] between the nearest-neighbor upsampled LR image and the HR image. Weber *et al.* [34] uses convolutional filters to preserve important visual details, and Hou *et al.* [9] recently proposed perceptual loss based method using deep learning.

However, a high similarity value does not imply good results when an image is restored to high resolution. Zhang *et al.* [40] proposed interpolation-dependent image downsampling (IDID) where given an interpolation method, the downsampled image that minimizes the sum of squared errors between the original input HR image and the obtained LR image interpolated to the input scale is obtained. Our method is most similar to IDID, but we mitigate its limitations

in that the upscaling process considers only simple interpolation methods and take full advantage of the recent advancements in deep learning-based SR.

3 Task-Aware Downscaling (TAD)

3.1 Formulation

We aim to study a *task-aware downscaled* (TAD) image that can be efficiently reconstructed to its original HR input. Let I^{TAD} denote our TAD image and I^{HR} as the original HR image. Our ultimate goal is to study the optimal downscaling function $g : I^{HR} \mapsto I^{TAD}$ with respect to the upscaling function f, which denotes our task of interest. The process of obtaining input I^{HR} is shown in the following equation:

$$I^{HR} = f(I^{TAD}) = f(g(I^{HR})).$$

The downscaling and upscaling functions g and f are both image-to-image mappings, and the input to g and the output of f are the same HR image I^{HR}. Thus, f and g are naturally modeled with a deep convolutional auto-encoder, each becoming the decoder and encoder part of the network.

Let θ_f and θ_g be the parameters of the convolutional decoder and encoder f and g, respectively. With the training dataset of N images $I_n^{HR}, n = 1, ..., N$ and L^{task} as the loss function that can differ task by task, our learning objective becomes:

$$\theta_f^*, \theta_g^* = \arg\min_{\theta_f, \theta_g} \frac{1}{N} \sum_{n=1}^{N} L^{task}\left(f_{\theta_f}\left(g_{\theta_g}\left(I_n^{HR}\right)\right), I_n^{HR}\right). \tag{1}$$

The desired I^{TAD} for downscaling and the reconstructed image I^{TAU} (task-aware upscaled image) can be calculated accordingly:

$$I^{TAD} = g_{\theta_g^*}\left(I^{HR}\right), \tag{2}$$

$$I^{TAU} = f_{\theta_f^*}\left(I^{TAD}\right). \tag{3}$$

3.2 Network Architecture and Training

In this section, we describe the network architecture and the training details. In this work, we mainly focus on the SR task and present SR-specific operations and confiugurations. The overall architecture is outlined in Figure 2.

Guidance image for better downscaling. In our framework, TAD images are obtained as the latent representation of the deep convolutional auto-encoder. However, without proper constraints, the latent representation may be arbitrary and does not look like the original HR image. Therefore, we propose a *guidance* image I^{guide}, which is basically a bicubic-downsampled LR image obtained from I^{HR}, to ensure visual similarity of our learned TAD image I^{TAD} with I^{HR}. The

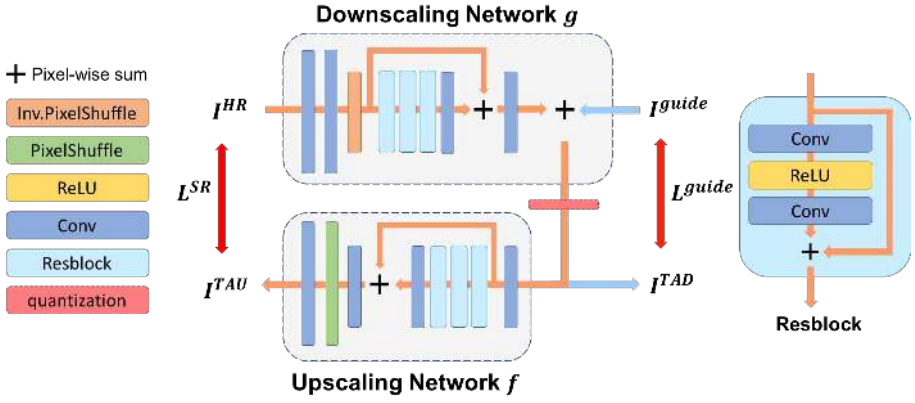

Fig. 2: Our convolutional auto-encoder architecture with three parts: downscaling network (g_{θ_g}, encoder), compression module, and upscaling network (f_{θ_f}, decoder). *Two* outputs, I^{TAD} and I^{TAU}, are obtained from Eqs. 2 and 3, and used to calculate the two loss terms in Eq. 4.

guidance image is used as a ground truth image to calculate the L1 loss with the predicted I^{TAD}. Incorporating I^{guide} and the new loss term, L^{guide}, changes the loss function in the original objective of Eq.1 to:

$$L^{task}\left(f(g(I_n^{HR})), I_n^{guide}, I_n^{HR}\right) = L^{SR}\left(f(I_n^{TAD}), I_n^{HR}\right) + \lambda L^{guide}\left(I_n^{TAD}, I_n^{guide}\right), \tag{4}$$

where L^{SR} is the standard L1 loss function for the SR task. θ_f and θ_g are omitted for the simplicity of notation. The hyperparameter λ is introduced to control the weights for the loss imposed by the guidance image w.r.t. the original SR loss. We can set the amount of trade-off between the reconstructed HR image quality and the LR TAD image quality by changing the value of λ. The effect of λ can be seen in Figure 4, and this will be analyzed more extensively in the experiment section.

Simple residual blocks as base networks. Our final deep convolutional auto-encoder model is composed of three parts: a downscaling network (encoder), a compression module, and an upscaling network (decoder). We jointly optimize all parts in an end-to-end manner, for the scaling factor of $\times 2$.

The encoder (g_{θ_g}) consists of a downscaling layer, three residual blocks, and a residual connection. The downscaling layer is a reverse version of sub-pixel convolution (also called pixel shuffle layer) [26], so that the feature channels are properly aligned and the number of channels is reduced by a factor of $\times 4$. We used two convolution layers with one ReLU activation for each residual block without batch normalization and bottleneck, which is the same as that used in EDSR [20]. Note that in our downscaling network g, the final output I^{TAD} is obtained by the addition of the output of the last conv. layer and the I^{guide} in a pixel-wise manner.

The decoder has almost the same simple architecture as the encoder, except the downscaling layer changes to the upscaling layer. The sub-pixel convolution layer [26] is used to upscale the output feature map by a factor of $\times 2$. Note that each scaling layer is located at the beginning (downscaling layer) and the end (upscaling layer) of the network to reduce the overall computational complexity of our model.

All our networks' convolution layers have a fixed channel size of 64, except for upscaling/downscaling layers, where we set the output activation map to have 64 channels. That is, for sub-pixel convolution with a scaling factor of $\times 2$, we first apply a 3×3 convolution layer to increase the number of channels to 256, and then align the pixels to reduce it again to 64.

Compression Module. Most deep networks have floating-point values for both feature activations and weights. Our TAD image output from the downscaling network is also represented with the default floating-point values. However, when displayed on a screen, most of the images are represented in true color (8 bits for each R, G, and B color channels). Considering that the objective of this work is to save a TAD image that is suitable for future application to SR, saving the obtained TAD image in RGB format is helpful for wider usage. We propose a compression module to achieve this goal.

A compression module is a structure for converting an image into a bitstream and storing it. We use a simple differentiable quantization layer that converts the floating-point values into 8-bit unsigned int (uint8) for this module. However, in the early iterations when the training is unstable, adding a quantization layer can result in training failure. Therefore, we omit it the layer until almost at the end of the training stage and insert our compression module again to fine-tune the network for a few hundred more iterations. The fine-tuned output TAD image then becomes a true-color RGB image that can be stored by lossless image compression methods, such as PNG. Although we used a single quantization layer for the compression module and saved the images in PNG format, this process can be generalized to the use of more complex image compression models as long as it is differentiable; thus, we call this part the *compression* module.

Multi-scale SR with extreme scaling factors. To deal with multiple scaling factors, we simply placed the original HR image in our downscaling model recursively, with minor changes in our architecture. Therefore, our model can (down)scale the HR image to the scaling factors of negative powers of 2. We even test our model with an extreme scaling factor of $\frac{1}{128}$ and show that our method can recover a reasonable $\times 128$ HR image from a tiny LR image. To the best of our knowledge, this work is the first to present the SR results for scaling factors of such an extreme level (over 16). Qualitative result and discussion can be seen in Figure 5.

Our architectural changes for multi-scale SR are as follows:

1. We omit the compression module during the recursive execution of the downscaling network, and replace the compression module of the final downscaling network to a simple rounding operation because a more beneficial alterna-

tive is to preserve the full information in floating-point values until the end where the final TAD image has to be saved.
2. The output of the downscaling network is modified to predict the guidance image itself directly by removing the pixelwise addtion of the guidance image.
3. During the recursive process, the network is fine-tuned for a few hundred iterations once every scaling factor of $\times 4$.

Upscaling the TAD image again requires the same recursive process, this time with the upscaling network. Although the exact downscaling and upscaling for our model, including recursive executions, are only for the scaling factors of powers of 2, combining our model with small-scale changes handled by a simple bicubic interpolation can still work. As shown in the experiments, this problem can be solved by applying a scale-invariant model, such as VDSR [13], to the obtained TAD image.

3.3 Extending to General Tensor Resizing Operations

Note that the goal of the SR task is to reconstruct the HR image I^{HR} from the corresponding LR image I^{LR}. Assuming I^{LR} (input low resolution image) with spatial size $H \times W$ and channels C, the upscaling function becomes $f : \mathbb{R}^{H \times W \times C} \mapsto \mathbb{R}^{sH \times sW \times C}$ where s denotes the scaling factor.

In this section, we formulate a generalized resizing operation, so that the proposed model can handle arbitrary resizing of an image tensor. Specifically, we consider the general *upscaling* task of $f : \mathbb{R}^{H \times W \times C} \mapsto \mathbb{R}^{sH \times rW \times tC}$, where s, r, and t are the scaling factors for the image height, width, and channels, respectively. $I^{HR} \in \mathbb{R}^{sH \times rW \times tC}$ is denoted again as a *high-resolution*[2]image tensor, and θ_f and θ_g are denoted as the parameters of our *new* models f_{θ_f} and g_{θ_g}, respectively. Training these models jointly with the same objective function of Eq. 1 completes our generalized formulation.

Note that if we constrain the scaling factor to $s = t = 1$, then the task becomes the image color space conversion. For example, if we consider the colorization task, the *downscaling* network g_{θ_g} performs a RGB to grayscale conversion where the spatial resolution is fixed and only the feature channel dimension is downsized. The *upscaling* network, f_{θ_f}, performs a colorization task. We use the similar model of a deep convolutional auto-encoder to obtain the TAD image I^{TAD}, which becomes a grayscale image that is optimal for the reconstruction of original RGB color image. For the colorization task, one major change in the network architecture is the removal of the downscaling layer in the encoder (g_{θ_g}) and the upscaling layer in the decoder (f_{θ_f}), because no spatial dimensionality change occurs in the color space conversions and the sub-convolution layers are not needed. Thus, the resulting network each has nine convolution layers. Other changes in the model configurations follow naturally: the guidance image I^{guide}

[2] We keep using the term *high-resolution* for the input tensor of its original scale, to have a consistent notation with the formulation in Sec. 3.1, although tensors in general don't use the word "resolution" to indicate its dimensions. Likewise, HR and LR image tensors represent the high-dimensional and the low-dimensional tensors.

becomes a grayscale image obtained using the conventional RGB to grayscale conversion method, and the task-aware *upscaled* image I^{TAU} becomes the colorized output image. For the compression module, a simple rounding scheme is used instead of a differentiable quantization layer.

4 Experiment

In this section, we report the results of our TAD model for SR (Sec. 4.1), analyze the results of our model thoroughly (Sec. 4.2), and apply our generalized model shown in Sec. 3.3 to the colorization task (Sec. 4.3).

4.1 TAD for Super-resolution

Datasets and evaluation metrics. We evaluate the performance on five widely used benchmark datasets: Set5 [3], Set14 [38], B100 [21], Urban100 [10], and the validation set of DIV2K [1]. All benchmark datasets are evaluated with scaling factors of $\times 2$ and $\times 4$ between LR and HR images. For the validation set of DIV2K that consists of 2K resolution images, we also perform experiments with extreme scaling factors of $\times 8$-$\times 128$. All the models we present in this section are trained on the 800 images from DIV2K training set [1]. No image overlap exists between our training set of images and the data we use for evaluation.

For the evaluation metric, we use PSNR to compare similarities between (1) the bicubic downscaled LR image and our predicted I^{TAD} (Eq. 2); and (2) the ground truth HR image and our predicted I^{TAU} (Eq. 3). To ensure a fair comparison with previous works, the input LR images of the reproduced SotA networks [13, 20] are downscaled by MATLAB's default `imresize` operation, which is implemented to perform bicubic downsampling with antialiasing. We apply the networks for both single channel (Y from YCbCr) and RGB color channel images. To obtain a single-channel image, an RGB color image is first converted to YCbCr color space, and the chroma channels (Cb, Cr) are discarded.

Comparison with the SotA. We compare our downscaling method **TAD** and upscaling method (**TAU**) with recent SotA models for single (VDSR [13]) and color (EDSR [20]) channel images. Since the single channel performance of EDSR+ and the color channel performance of VDSR are not provided in the reference papers, we reproduced them for the comparison. For *VDSR and *EDSR+ under TAD as the downscaling method, we re-train the reproduced networks using TAD-HR image pairs, instead of conventional LR-HR pairs for bicubic-downsampled LR images. Quantitative evaluations are summarized in Table 1.

The results show that our jointly trained TAD-TAU for the color image SR outperforms all previous methods in all datasets. Moreover, EDSR+ trained with TAD-HR images (*down- and up-scaling* ***not*** *jointly trained as an auto-encoder*) boosts reconstruction performance considerably, gaining over 5 dB additional PSNR in some benchmarks. The same situation holds for the single channel settings. The TAU network architecture is much more efficient (comprising 10

Table 1: Quantitative PSNR (dB) results on benchmark datasets: Set5, Set14, B100, Urban100, and DIV2K. The red color indicates the best performance, and the blue color indicates the second best. (*: reproduced performance)

		Single Channel Results / Color Channel Results					
Downscaling		Bicubic			**TAD**(Ours)		
Upscaling		TAU(baseline)	VDSR [13]	EDSR+ [20]	TAU	*VDSR	*EDSR+
Set5	×2	35.84/36.04	37.53/35.08	37.95/36.09	37.69/38.46	37.68/38.76	37.98/39.44
	×4	31.20/29.52	31.35/29.39	32.17/30.71	31.59/31.81	31.60/31.96	32.36/32.49
Set14	×2	32.89/30.99	33.03/30.93	33.65/31.97	33.90/35.52	33.88/35.92	34.07/36.58
	×4	27.92/26.28	28.01/26.26	28.50/27.14	28.36/28.63	28.38/28.76	28.82/29.24
B100	×2	31.74/30.40	31.90/30.42	32.22/31.40	32.62/36.68	32.65/36.87	32.83/37.59
	×4	27.20/25.88	27.29/25.87	27.54/26.45	27.57/28.51	27.57/28.53	27.86/28.97
Urban100	×2	30.64/29.13	30.76/29.19	32.51/31.47	31.96/35.03	32.16/35.50	32.86/35.55
	×4	25.08/23.66	25.18/23.68	26.25/25.34	25.56/26.63	25.66/26.98	26.50/27.76
DIV2K	×2	35.17/33.91	35.29/33.79	35.91/35.12	36.13/39.01	36.18/39.42	36.52/40.21
	×4	29.73/28.40	29.63/28.31	30.29/29.38	30.25/31.16	30.25/31.34	30.73/31.88

convolution layers) than the compared networks, VDSR (20 convolution layers) and EDSR+ (68 convolution layers).

The qualitative results in Figure 3 show that only TAU for the color image perfectly reconstructs the word, "presentations". TAU for the single-channel image also provides clearer characters than the previous SotA methods.

Training details. We trained all models with a GeForce GTX 1080 Ti GPU using 800 images from the DIV2K training data [1]. For both training and testing, we first crop the input HR images from the upper and left sides so that the height and width of the image are divisible by the scaling factors. Then, we obtain the guidance images (single channel or color channel LR images with regard to the experiment setting) by using MATLAB `imresize` command. We randomly crop 16 patches of 96×96 HR sub images, with each patch coming from a different HR image, to construct the training mini-batch. Our downscaling and upscaling networks are fully convolutional and can handle images of arbitrary size. We normalized the range of the input pixel values to [-0.5,0.5] and output pixel values to [0,1], and the L1 loss is calculated to be in the range of [0,1]. To optimize our network, we use the ADAM [15] optimizer with $\beta_1 = 0.9$. The network parameters are updated with a learning rate of 10^{-4} for 3×10^5 iterations.

4.2 Analysis

In this section, we perform two experiments to improve understanding of our TAD model and discuss the results.

Investigating LR-HR image quality trade-off. The objective for training our model is given in Sec. 3.1, Eq. 4. The hyperparameter λ controls the weight

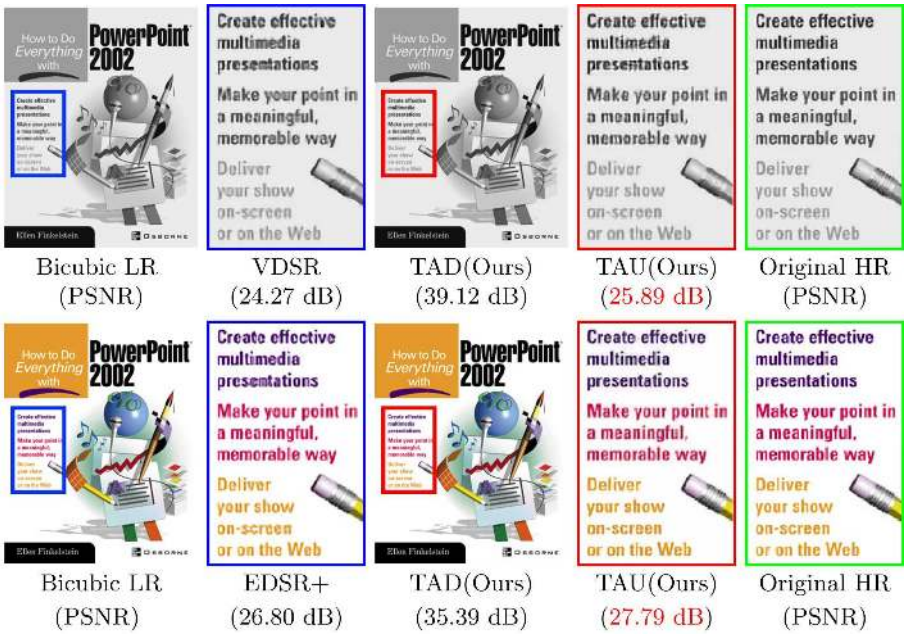

Fig. 3: Qualitative SR results of "ppt3"(Set14). The top and bottom rows show the results for single (Y) and color (RGB) channel images, respectively. In both gray and color images, TAD produces more decent LR images compared with Bicubic and guarantees much better HR reconstructions when upscaled with TAU. This figure is best viewed in color, and by zooming into the electronic copy. The scaling factor is $\times 2$.

between two loss terms: L^{SR} for HR image reconstruction and L^{guide} for LR image guidance. If $\lambda = 0$, then our framework becomes a simple deep convolutional auto-encoder model for the task of SR, without any constraint in producing a high quality downscaled image. Conversely, if $\lambda = \infty$, L^{SR} is ignored, then and our framework becomes a downscaling CNN with ground truth downscaling method as bicubic downsampling. In this study, we explore the effect of the influence of guidance image I^{guide}, and find that changing the weight λ allows us to control the quality of generated HR (I^{TAU}) and LR (I^{TAD}) images. This effect is visualized in Figure 4.

We train our TAD model for the scaling factor of $\times 2$, first with $\lambda = 0$ and gradually increase its value up to 10^2. For each λ, we measure the average PSNR for 10 validation images of DIV2K [1] and plot the values , as shown in the top-left corner of Figure 4. We chose $\lambda = 10^{-1}$ where the PSNR for HR images (39.81 dB) and LR images (40.69 dB) are similar, as the default value for our model and use it throughout all the SR experiments. The compression module is not used for this experiment. The exact PSNR accuracy for different values of λ will be reported in the supplementary materials due to the space limit.

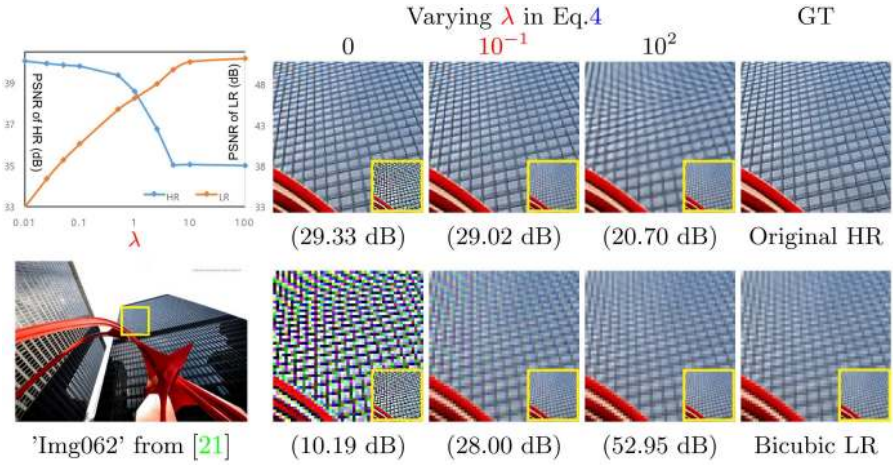

Fig. 4: TAD-TAU reconstruction performance trade-off. Smaller values of λ give a high upscaling performance with noisy TAD image. We choose λ from the intersection of the curves, where both TAD/TAU images give satisfactory results. PSNR for LR image is measured with bicubic-downsampled image, and for HR images with the original GT.

Multi-scale extreme SR. The results of recursive multi-scale SR operation with extreme scaling factors described in Sec. 3.2 are shown in Figure 5. In this experiment, the last conv. of our downscaling network predicts TAD images directly. As the guidance image for each scaling factors is not needed to produce TAD/TAU images, it improves practical applicability of our model. Quantitative analysis and more of qualitative results will be provided in the supplementary materials due to the page limit.

Runtime analysis. Our model efficiently achieves near real-time performance while still maintaining SotA SR accuracy. Each of our scaling network consists of 10 convolution layers and one sub-convolution (pixel shuffle) layer, and a full HD image (1920×1080) can be upscaled in 0.14s with a single GeForce GTX 1080 GPU Ti. Our model clearly has a major advantage over the recent EDSR+ (70.88s), which is a heavy model with 68 convolution layers.

4.3 Extension: TAD for Colorization

We follow the exact formulation described in Sec. 3.3 and perform the color space conversion experiments accordingly. All experiments use the DIV2K training image dataset [1] for training, and B100 and Urban100 datasets for evaluation. We use a single Y channel image from YCbCr color space as I^{guide}, and we choose our hyper-parameter $\lambda = 5$ to place a strong constraint on our TAD image.

To demonstrate the effectiveness of our proposed framework, we train another image colorization network that has the same architecture as our upscaling net-

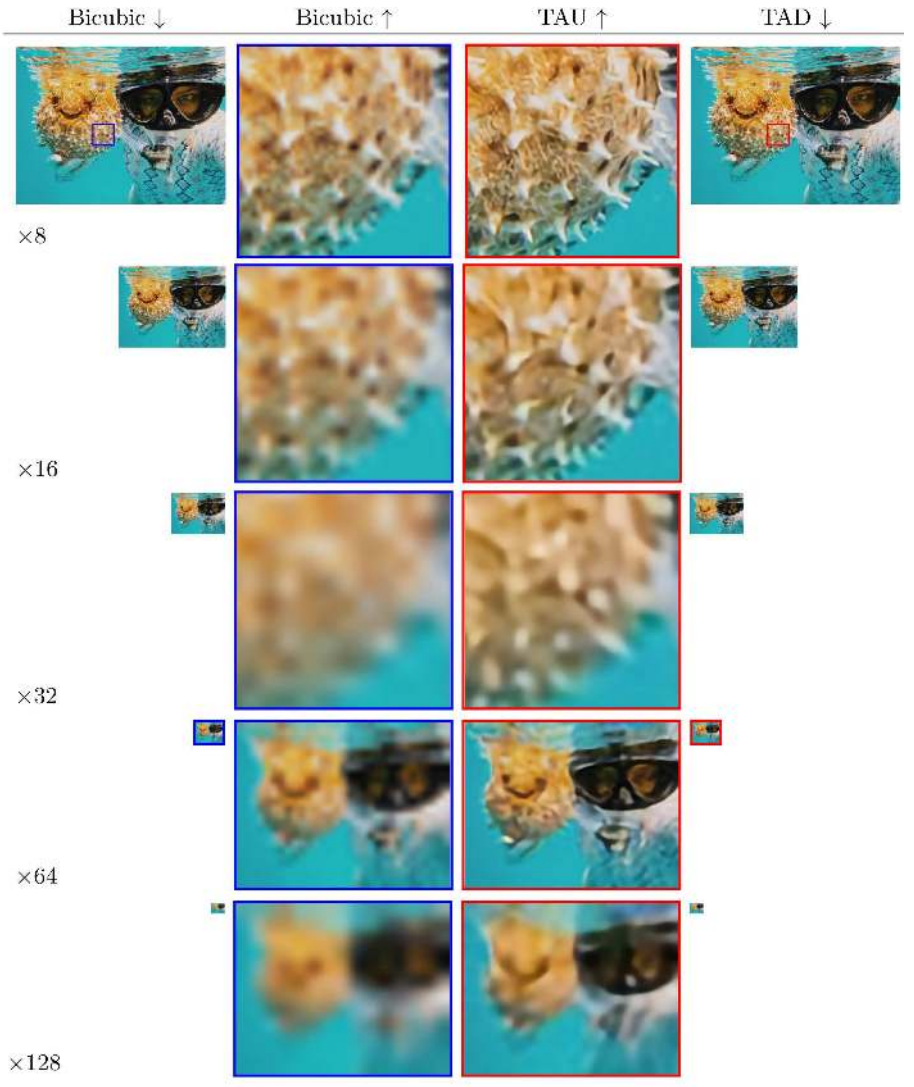

Fig. 5: Results of extreme scaling factors up to ×128. Our TAD images over all scales have decent visual quality with respect to Bicubic↓, and our TAU images are much cleaner and sharper than those of Bicubic ↑. All resized results are produced by **a single joint network of TAU and TAD** (Figure 2), with a scaling factor of ×2. Considering that the ×64 and ×128 downscaled images have only 31×24 and 15×12 pixels respectively, we visualize the full image for these extreme scaling factors. The generated I^{TAU} is downscaled again - with Bicubic↓ - for visualization. Note the detailed recovery of the spines of the pufferfish in ×8 and a surprisingly realistic global structures reconstructed in ×64.

Fig. 6: Qualitative image colorization results. The leftmost image is used as I^{guide} for our model and input grayscale for the baseline. The channel scale factor $\times 3$.

work with conventional grayscale-HR image pairs. The results in Figure 6 show that the colorization network trained in a standard way clearly cannot resolve the color ambiguities, whereas our TAD Gray image contains the necessary information for restoring original pleasing colors as demonstrated in the reconstructed TAD color. Quantitatively, while the baseline model achieves an average PSNR of 24.21 dB (B100) and 23.29 dB (Urban100), our model outputs much higher performance values of 36.14 dB (B100) and 33.68 dB (Urban100).

The results clearly demostrate that the TAD-TAU framework is also practically very useful for both the color to gray conversion and gray to color conversion (colorization) tasks.

5 Conclusion

In this work, we present a novel *task-aware* image downscaling method using a deep convolutional auto-encoder. By jointly training the downscaling and upscaling processes, our task-aware downscaling framework greatly alleviates the difficulties in solving highly ill-posed resizing problems such as image SR. We have shown that our upscaling method outperforms previous works in SR by a large margin, and our downscaled image also aids the existing methods to achieve much higher accuracy. Moreover, valid scaling results with extreme scaling factors are provided for the first time. We have demonstrated how our method can be generalized and verified our framework's capability in image color space conversion. Apart from the tasks examined in this study, we believe that our approach provides a useful framework for handling images of various sizes. Promising future work may include deep learning based image compression.

6 Acknowledgement

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government(MSIT) (No. NRF-2017R1A2B2011862)

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: CVPRW (2017)
2. Allebach, J., Wong, P.W.: Edge-directed interpolation. In: ICIP (1996)
3. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC (2012)
4. Chang, H., Yeung, D.Y., Xiong, Y.: Super-resolution through neighbor embedding. In: CVPR (2004)
5. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV (2014)
6. Gao, X., Zhang, K., Tao, D., Li, X.: Image super-resolution with sparse neighbor embedding. IEEE Transactions on Image Processing **21**(7), 3194–3205 (2012)
7. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: ICCV (2009)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
9. Hou, X., Duan, J., Qiu, G.: Deep feature consistent deep image transformations: Downscaling, decolorization and HDR tone mapping. arXiv:1707.09482 (2017)
10. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR (2015)
11. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
12. Keys, R.G.: Cubic convolution interpolation for digital image processing. IEEE Transactions on Acoustics, Speech, and Signal Processing pp. 1153–1160 (1981)
13. Kim, J., Lee, J., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
14. Kim, K.I., Kwon, Y.: Single-image super-resolution using sparse regression and natural image prior. TPAMI **32**(6), 1127–1133 (2010)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. ICLR (2015)
16. Kopf, J., Shamir, A., Peers, P.: Content-adaptive image downscaling. ACM Transactions on Graphics **32**(6), 173 (2013)
17. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR (2017)
18. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
19. Li, X., Orchard, M.T.: New edge-directed interpolation. IEEE Transactions on Image Processing **10**(10), 1521–1527 (2001)
20. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPRW (2017)
21. Martin, D.R., Fowlkes, C.C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
22. Michaeli, T., Irani, M.: Nonparametric blind super-resolution. In: ICCV (2013)
23. Mitchell, D.P., Netravali, A.N.: Reconstruction filters in computer-graphics. In: SIGGRAPH. pp. 221–228 (1988)
24. Öztireli, A.C., Gross, M.: Perceptually based downscaling of images. ACM Transactions on Graphics **34**(4), 77:1–77:10 (2015)

25. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
26. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR (2016)
27. Shocher, A., Cohen, N., Irani, M.: "Zero-Shot" Super-resolution using deep internal learning. In: CVPR (2018)
28. Sun, J., Xu, Z., Shum, H.Y.: Image super-resolution using gradient profile prior. In: CVPR (2008)
29. Tai, Y.W., Liu, S., Brown, M.S., Lin, S.: Super resolution using edge prior and single image detail synthesis. In: CVPR (2010)
30. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L., Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., et al.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: CVPRW (2017)
31. Timofte, R., Smet, V.D., Gool, L.J.V.: A+: adjusted anchored neighborhood regression for fast super-resolution. In: ACCV (2014)
32. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: CVPR (2018)
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error measurement to structural similarity. IEEE Transactions on Image Processing **13**, 600–612 (2004)
34. Weber, N., Waechter, M., Amend, S.C., Guthe, S., Goesele, M.: Rapid, detail-preserving image downscaling. ACM Transactions on Graphics **35**(6), 205:1–205:6 (2016)
35. Yang, J., Wang, Z., Lin, Z., Cohen, S., Huang, T.: Coupled dictionary training for image super-resolution. IEEE Transactions on Image Processing **21**(8), 3467–3478 (2012)
36. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE Transactions on Image Processing **19**(11), 2861–2873 (2010)
37. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Proceedings of the International Conference on Curves and Surfaces (2010)
38. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Proceedings of the International Conference on Curves and Surfaces (2010)
39. Zhang, L., Wu, X.: An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Transactions on Image Processing **15**(8), 2226–2238 (2006)
40. Zhang, Y., Zhao, D., Zhang, J., Xiong, R., Gao, W.: Interpolation-dependent image downsampling. IEEE Transactions on Image Processing **20**(11), 3291–3296 (2011)